



## Desafío: Operaciones de modificación y exclusión en el ProductoDAO

Utilizando todo lo que aprendimos en la refactorización de la lógica de acceso a la capa de datos del `ProductoController` para la clase `ProductoDAO` es el momento de poner en práctica lo que aprendimos para refactorizar las operaciones de modificación y exclusión de productos siguiendo las buenas prácticas que aprendimos.



### Opinión del instructor

Empecemos por el método de `eliminar` de la clase `ProductoController`. Lo que vamos hacer primero es crear una llamada a `productoDao.eliminar(id);`. Este método aún no existe, entonces vamos a crearlo en el `ProductoDAO` y mover la lógica de exclusión del producto para allá. Vamos a dejarla más sencilla también, removiendo pedazos de código que ya no son más necesarios, como la de tomar una conexión.

Recordemos que no debemos más estar lanzando la `SQLException` para las demás capas. Los errores que una clase puede lanzar deben ser tratados en ella de forma que podamos lanzar, como máximo, una excepción del tipo *unchecked*.

Por último vamos a ajustar el código referente a la exclusión de un producto en la clase `ControlDeStockFrame` y listo. Podemos hacer lo mismo con el método de modificación.

Así va a quedar el resultado:

```
// ControlDeStockFrame

private void modificar() {
    if (tieneFilaElegida()) {
        JOptionPane.showMessageDialog(this, "Por favor, elije una fila para modificar.");
        return;
    }

    Optional.ofNullable(modelo.getValueAt(tabla.getSelectedRow(), 0))
        .ifPresentOrElse(fila -> {
            Integer id = Integer.valueOf(modelo.getValueAt(tabla.getSelectedRow(), 1));
            String nombre = (String) modelo.getValueAt(tabla.getSelectedRow(), 2);
            String descripcion = (String) modelo.getValueAt(tabla.getSelectedRow(), 3);
            Integer cantidad = Integer.valueOf(modelo.getValueAt(tabla.getSelectedRow(), 4));

            var filasModificadas = this.productoController.modificar(id, nombre, descripcion, cantidad);

            JOptionPane.showMessageDialog(this, String.format("Se modificó el producto %s", nombre));
        }, () -> JOptionPane.showMessageDialog(this, "Por favor, elije una fila para modificar."));
}

private void eliminar() {
    if (tieneFilaElegida()) {
        JOptionPane.showMessageDialog(this, "Por favor, elije una fila para eliminar.");
        return;
    }

    Optional.ofNullable(modelo.getValueAt(tabla.getSelectedRow(), 0))
        .ifPresentOrElse(fila -> {
            Integer id = Integer.valueOf(modelo.getValueAt(tabla.getSelectedRow(), 1));

            var filasModificadas = this.productoController.eliminar(id);

            modelo.removeRow(tabla.getSelectedRow());
            JOptionPane.showMessageDialog(this, "Se eliminó el producto con ID " + id);
        }, () -> JOptionPane.showMessageDialog(this, "Por favor, elije una fila para eliminar."));
}
```

```
        String.format("%d item eliminado con éxito", cantidad), () -> JOptionPane.showMessageDialog(this, "Por favor, reinicie la aplicación para ver los cambios.")
    }

// ProductoController
public int modificar(String nombre, String descripcion, Integer cantidad) {
    return productoDao.modificar(nombre, descripcion, cantidad);
}

public int eliminar(Integer id) {
    return productoDao.eliminar(id);
}

// ProductoDAO
public int eliminar(Integer id) {
    try {
        final PreparedStatement statement = con.prepareStatement("DELETE FROM producto WHERE id = ?");

        try (statement) {
            statement.setInt(1, id);
            statement.execute();

            int updateCount = statement.getUpdateCount();

            return updateCount;
        }
    } catch (SQLException e) {
        throw new RuntimeException(e);
    }
}

public int modificar(String nombre, String descripcion, Integer cantidad) {
    try {
```

```
final PreparedStatement statement = con.prepareStatement(
    "UPDATE PRODUCTO SET "
    + " NOMBRE = ?, "
    + " DESCRIPCION = ?, "
    + " CANTIDAD = ?"
    + " WHERE ID = ?");

try (statement) {
    statement.setString(1, nombre);
    statement.setString(2, descripcion);
    statement.setInt(3, cantidad);
    statement.setInt(4, id);
    statement.execute();

    int updateCount = statement.getUpdateCount();

    return updateCount;
} catch (SQLException e) {
    throw new RuntimeException(e);
}
```

[COPIA EL CÓDIGO](#)